What is claimed is:

Sub A1>

1. A system, comprising:

a data source, having a plurality of different lines;

a plurality of programmable delay elements, each coupled to one of said plurality of lines, to control a delay in said one of said lines to produce delayed values; and

a register, storing values for said programmable delay elements which respectively control an amount of delay caused by said delay elements.

2. A system as in claim 1, wherein there are one of said programmable delay elements for each of said plurality of lines.

3. A system as in claim 1, wherein said register stores a plurality of values, each of said plurality of values controlling one of said programmable delay elements.

4. A system as in claim 3, further comprising a non volatile memory, storing said plurality of values.

5. A system as in claim 1, further comprising an arbitration logic, coupled to said plurality of delayed

values, and operating to determine relative timing of said plurality of lines.

6. A system as in claim 5, wherein said arbitration logic includes a first element which produces a set of first values for said register, and a second element which determines relative arrival of signals based on said first values.

7. A system as in claim 6, wherein said arbitration logic dithers between different sets of values, and determines which of said plurality of values produces best desired result, and stores said best result.

Sub A1

8. A system as in claim 1, further comprising a graphics device, and wherein said signals are from said graphics device.

9. A system as in claim 1, further comprising a non-volatile memory, storing values for said delay elements, and loading said values into said register at a specified time.

10. A system as in claim 8, further comprising arbitration logic, coupled to said plurality of delayed

values, dithers between different sets of values, determines which of said plurality of values produces a best desired result, and stores said best result in said non-volatile memory.

Sub A1

11. A system as in claim 10, wherein said best result is one where the plurality of delayed signals are received at substantially the same time.

12. A system as in claim 10, wherein said best result is one where the plurality of delayed signals are received at a time skew that allows certain logic elements to operate correctly.

13. An electronic device, comprising:

a first device producing a plurality of first outputs;

a plurality of programmable delay elements, each of said plurality of programmable delay elements connected at one end to one of said plurality of outputs and each producing a second output which is delayed relative to said first outputs;

a levelization register, storing a plurality of values, said values each individually controlling one of said delay elements to control an amount of delay caused by said delay element to one of said plurality of outputs.

14. A device as in claim 13, further comprising arbitration logic, connected to each of said second outputs, and determining a relative delay among said second outputs.

15. A device as in claim 14, wherein said arbitration logic is responsive to a system event flag, which indicates a specified event in the system.

16. A device as in claim 15, wherein said specified event is a hardware change.

17. A device as in claim 15, wherein said specified event is a system crash.

18. A system as in claim 15, wherein said arbitration logic is responsive to said flag to produce a first set of values for said levelization register, command said first device to produce said signals, and determine a relative delay among said signals based on said first set of values.

19. A system as in claim 14, further comprising a non volatile memory which stores levelization values, said

arbitration logic storing said levelization values in said non volatile memory.

Sub A1

20. A system as in claim 15, further comprising, at initial system start up, downloading values from said non volatile memory to said levelization register.

21. A system as in claim 15, wherein said programmable delay elements are phase locked loops.

22. A method comprising:

receiving a plurality of signals from an external device, each of said plurality of signals related to each other; and

programmably delaying some of said signals relative to others of said signals according to prestored values.

23. A method as in claim 22, wherein plurality of signals are signals from a bus.

24. A method as in claim 23, wherein said plurality of signals are signals from a graphics bus.

25. A method as in claim 22, wherein said delaying comprises storing delay values in a non volatile memory; and

using said values in said non volatile memory to adjust a value of a programmable delay element.

26 25. A method as in claim 24, further comprising determining if a system event has occurred, and storing new delay values in said non volatile memory responsive to said system event occurring.

27 26. A method as in claim 25, wherein said system event is a change in system hardware configuration.

28 27. A method as in claim 25, wherein said system event is a system crash.

29 28. A method as in claim 25, wherein said reobtaining comprises dithering values in a register that stores values for said programmable delay, determining results, and accepting values which have produced a specified delay.

30 29. A method as in claim 28, further comprising storing said values in said non-volatile memory.

DMB

31 ~~30~~. A method as in claim ~~28~~ 29, wherein said specified delay is a result where there is a minimal delay between arrival of all signals.

Sub A1

DMB

~~31~~ 32. A method as in claim ~~28~~ 29, wherein said specified delay is a result where there is a specified delay between arrival of all signals which allows for clock skew in at least one specified logic element.

DMB

~~32~~ 33. A method of equalizing time delays of signals, comprising:

providing a plurality of signals which are produced in times that are synchronized with one another;

delaying each of said plurality of signals by a respective amount, wherein each of said respective amounts is different then each other respective amount for a different one of said signals;

testing said signals, to determine relative amounts of delays in said signals, to produce said delay amount; and

using said delay amounts to delay said signals.

DMB

~~33~~ 34. A method as in claim ~~32~~ 33, wherein said plurality of signals are signals from a graphics processing device.

DMB

35 34. A method as in claim 32, wherein said delaying comprises delaying each of the signals by respective amounts which causes them to arrive at a specified location at substantially similar times.

Sub #1

36 35. A method as in claim 32, wherein said delaying comprises delaying said signals by specified amounts which causes them to arrive at said location at specified times which are skewed relative to one another, wherein said skew is related to a clock margin of a system.

DMB

37 36. A method of setting delays in a system, comprising:

storing values indicative of time delays in a register, said time delays representing delays to be applied to signals to obtain a specified result;

detecting a system event which indicates that said time delays should be changed;

when said event is not detected, using said values in said register to cause signal delays, by applying said values to respective programmable delay elements; and

when said event is detected, using a logic element to determine new delay values and applying said new delay values to said programmable delay elements to cause signal delays based on said new delay values.

DMB

38 37. A method as in claim 36, wherein said using comprises applying a plurality of delay values to a plurality of respective programmable delay elements to thereby delay a plurality of lines.

Sub #1

39 37
DMB 39 38. A method as in claim 36, further comprising storing said new delay values in a non-volatile memory.

40 37
DMB 39. A method as in claim 36, wherein said system event includes a change of system components.

41 37
DMB 40. A method as in claim 36, wherein said system event is an operating system crash.

DMB 42
41. A method, comprising:

receiving a plurality of signals in parallel from a specified device, which signals are produced at substantially synchronized times;

applying said plurality of said signals to programmable delay devices which allow individual delay of said signals; and

controlling said plurality of programmable delay devices to allow the signals to arrive to at least one specified location in a specified way.

43 42. A method as in claim 41, wherein said specified way is that said signals are substantially synchronized with one another at said at least one specified location.

44 43. A method as in claim 41, wherein said specified way is that said signals are have a specified relationship to one another at said at least one specified location, which specified relationship is not synchronized, to allow a system to work properly.

45 44. A method as in claim 41, further comprising storing said information in a non-volatile memory.